Remarks

As stated above, Applicants appreciate the Examiner's thorough examination of the subject application and request reexamination and reconsideration of the subject application in view of the preceding amendments and the following remarks.

As of the office action of June 5, 2009, claims 4-6, 8-14, 20, 22, and 24-27 were pending in the subject application, of which claims 4, 20, 22, and 24-26are independent claims. With this response applicant has amended claims 4, 6, 8, 12, 13, 20, 22, and 24-26. Applicant has also canceled claims 15-18 and added new claim 28.

A. Amendments to the Specification

Applicant has amended the specification by adding a "RELATED APPLICATIONS" section. This section explicitly calls out the PCT and French applications upon which this U.S. application is based.

Applicant has also amended the specification by adding the terminology "computer readable storage medium." No new matter has been added with this amendment because (A) a person of ordinary skill in the art would recognize computer readable storage medium as a well known term used to describe memories, hard drives, and the like, and (B) the new terminology is merely nomenclature for matter already included in the specification; it merely provides a label for subject matter already disclosed. (*See* M.P.E.P § 608.01(o) "while an applicant is not limited to the nomenclature used in the application as filed, he or she should make appropriate amendment of the specification whenever this nomenclature is departed from by amendment of the claims so as to have clear support or antecedent basis in the specification").

Applicant has also amended the specification by replacing the term "Nbep = Nopi" with the term "Nbep $\geq$ NOpi." No new matter has been added with this amendment because the

"equal sign" in the specification of the US application is a typographical error introduced during translation of the international application. Page 31 of the specification in the French-language international-stage application contains the term "Nbep ≥ NOpi." *See* PCT/FR00/02349 at specification, page 31. The term "Nbep ≥ NOpi" also appears in FIG. 3h, block 400.

B. Restriction Requirement

The Examiner required Applicant to elect between two sets of claims: Group I (consisting of claims 4-6, 8-14, 20, 22, and 24-28), and Group II (consisting of claims 15-18). *See* Restriction Requirement mailed Jan. 6, 2009. Applicant provisionally elected Group I with traverse in a response dated Mar. 6, 2009. In the most recent office action, the Examiner made the restriction final. In response to the finality of the restriction, Applicant elects Group I and has canceled claims 15-18.

C. New Claims

Applicant added new claim 28. No new matter has been added with claim 28 because it contains the subject matter originally found in claim 7. Original claim 7 was canceled in an earlier action.

D. Rejections Under 35 U.S.C. § 101

The Examiner rejected claim 22 because it contains the term "working memory," which, according to the Examiner, is not readily known in the art. Office Action at pp. 3-4. In response, Applicant has amended claim 22 by removing the word "working," thus replacing the term "working memory" with the term "memory." Applicant contends that the term "memory" is readily known in the art. Accordingly, Applicant requests withdrawal of the § 101 rejection of claim 22.

E. Rejections Under 35 U.S.C. § 112

The Examiner rejected claims 4, 8, 12, 13, 20, and 22 under 35 U.S.C. § 112, second paragraph. Id. at p. 4. The Examiner stated that the claims are "indefinite for failing to . . . distinctly claim the subject matter[.]" Id. In response to the Examiner's rejections, Applicants have amended the claims as discussed below.

Regarding claim 4, the Examiner rejected claim 4 stating that "it is indefinite whether 'said stack of variable types' refers to the execution stack or the type stack. Id. Applicant has amended claim 4, replacing the term "said stack" with the term "said type stack." Applicant requests withdrawal of the § 112 rejection of claim 4 since the term now unambiguously refers to the "type stack."

Regarding claim 6, the Examiner stated that "it is unclear what Applicant is attempting to claim in claim 6." Id. Applicant notes that the Examiner has not stated whether claim 6 is rejected under § 112. However, Applicant has amended claim 6 so that the claim is directed toward "variable types" obeying subtype relationships. Applicant requests withdrawal of the § 112 rejection of claim 6.

Regarding claim 8, the Examiner stated that "it is indefinite whether 'said stack of variable types' refers to the execution stack or the type stack." Id. Applicant has amended claim 8 replacing the term "'said stack of variable types " with the term "said type stack." Applicant requests withdrawal of the § 112 rejection of claim 8 since the term now unambiguously recites "type stack."

Regarding claim 12, the Examiner stated that "it is indefinite whether 'said type execution stack' refers to the execution stack or the type stack." Id. Applicant has amended claim 12 replacing the term "said type execution stack" with the term "said type stack." Applicant

requests withdrawal of the § 112 rejection of claim 12 since the term now unambiguously recites "type stack."

Regarding claim 13, the Examiner stated that "it is unclear if [the term "the stack"] is a reference to "said type stack" in the same clause." Id. Applicant has amended claim 13 replacing the term "the stack" with the term "said type stack." Applicant requests withdrawal of the § 112 rejection of claim 13 since the term now unambiguously recites "type stack."

Regarding claim 20, the Examiner stated that "it is indefinite whether 'said stack is empty' refers to the type stack." Id. at p. 5 Applicant has amended claim 20 replacing the term "said stack" with the term "said type stack." Applicant requests withdrawal of the § 112 rejection of claim 20 since the term now unambiguously recites "type stack."

Regarding claim 22, the Examiner stated that "it is indefinite whether said stack' refers to the execution stack or the type of stack variables." Id. at p. 5 Applicant has amended claim 22 replacing the term "said data type of said stack" with the term "a data type of said type stack." Applicant requests withdrawal of the § 112 rejection of claim 22 since the term now unambiguously recites "a data type of said type stack."

F. Rejections Under 35 U.S.C. § 103

The Examiner rejected claims 4, 8-14, 20, 22, and 24-27 under 35 U.S.C. § 103 over U.S. Patent No. 5,740,441 ("Yellin") in view of U.S. Patent No. 6,071,317 ("Nagel"). Id. Applicant respectfully traverses the rejection because the combination of Yellin and Nagel does not appear to disclose or suggest each and every element of the claimed invention. Therefore, the combination of Yellin and Nagel does not result in the claimed invention.

Claim 4

Yellin and Nagel do not appear to disclose each and every element of claim 4. Yellin appears to disclose a method of verifying a program fragment (...) consisting of an object code and including at least one subprogram consisting of a series of instructions manipulating (...) operand registers. Further, Yellin appears to disclose, for each subprogram, initializing a type stack and a table of register types through data representing a state of said virtual machine on initialization of an execution of said temporarily stored object code (col. 8, l. 41–44). However, many of the elements of the independent claims do not appear to be disclosed, taught, or suggested in Yellin.

For example, Yellin does not appear to disclose "said current instruction being a target of a branching instruction, said verification process including verifying that said type stack is empty and rejecting said program fragment otherwise," as claimed in claim 4.

Yellin also appears to disclose checking for stack overflow and underflow. (Yellin at col. 9, l. 30–39, col 9, l. 61 to col. 10, l. 17). However, Yellin does not appear to disclose or suggest "verifying that the stack is empty if the current instruction is the target of a branching instruction." This condition, which is part of criterion C3 (specification, p. 30, l. 22-23), alleviates the need to merge stack description across branching points (as taught e.g. by Yellin, col. 11, l. 38 to col. 12, l. 53), thus reducing the requirement for larger amounts of memory than are typically available on embedded systems that are a target of the present invention (see specification, p. 5, l. 20 to p. 6, l. 7). Yellin also does not appear to teach or suggest verifying that said type stack is empty, as claimed.

Furthermore, Yellin also does not appear to disclose "said verification process being successful when said table of register types is not modified in the course of a verification", i.e.,

that the termination condition for the verification process is that the table of register types are *left unchanged* over the course of a verification of all the instructions. The termination condition taught by Yellin is that there is no instruction whose changed bit is set, not that the table of register types is left unchanged (col. 9, l. 5–8). The changed bit of an instruction is set if the merging of either the register table or the type stack of a predecessor instruction causes a change (col. 12, l. 40–45). Hence the termination taught by Yellin differs from Applicant's in that Yellin's considers both registers and stack while Applicant's only considers registers. Furthermore, in relation with registers, Yellin teaches the use of one "SnapShot" virtual register array per instruction (col. 7, l. 40–43), and the verification of an instruction may affect the virtual register array of other instructions (its successors) (col.12 , l. 20–35 and col. 12, l. 40–45). Hence Yellin's termination condition rests upon the absence of a modification of multiple register tables, while Applicant's rests upon a single register table. For these reasons, Yellin does not disclose appear to disclose the elements of claim 4.

Nagel does not appear to remedy Yellin's deficiencies. Nagel generally discloses a method for modifying program executable code, possibly on an embedded system. Accordingly, Applicants contend that combining Nagel's disclosure with Yellin's would not have taught one of ordinary skill in the art the method of claim 4.

Since Yellin and Nagel do not appear to disclose or suggest each and every element of claim 4, Applicant respectfully requests withdrawal of the § 103 rejection of claim 4, and claims 8-14 because they are dependent upon claim 4.

Claim 22

Yellin and Nagel do not appear to disclose each and every element of claim 22. The Examiner cites Yellin (col. 10, l. 53–60) and (col. 10, l. 43–48) in the rejection of claim 22.

Yellin discusses verifying the validity of register numbers that instructions write to (col. 10, l. 53–60). Yellin also appears to discuss continuing the verification if reads from registers are valid (col. 10, l. 43–48). However, these passages to not appear to disclose or suggest the elements of the claims.

Yellin discloses annotating each current instruction with said data type of said type stack before and after execution of said current instruction, with an annotation data being calculated by means of analysis of the data stream relating to said current instruction (col. 9, l. 39–52; col. 10, l. 6–16).

Yellin also discusses detecting (…) the existence of branchings (col. 11, l. 9–25). Yellin discloses detecting (…) the existence of branching targets (col. 10, l. 66–67). Yellin discloses detecting that the execution stack is not empty, said detecting operation being carried out on the basis of said annotation data of said type of stack variables allocated to each current instruction (col. 9, l. 30–37). However, this latter teaching appears to be directed to detecting a potential stack underflow, and Yellin does not appear to suggest any relation with the detection of the *existence of branchings or of branching targets*, as claimed. Therefore Yellin cannot be said to disclose detecting (…) the existence of branchings, or respectively of branching targets, for which said execution stack is not empty.

Yellin discloses marking registers accessed or modified by instructions that are the target of a jsr instruction (col. 10, l. 61 to col. 11, l. 6). Yellin also discloses determining the successor instructions of a current instruction (col. 11, l. 7–30). However, Yellin does not appear to disclose any transformation of the instructions that are to be verified. Yellin does not disclose to empty contents of said execution stack into temporary registers before said branching and to reestablish said execution stack from said temporary registers after said branching, which is a

method for complying with requirement C3 mentioned above. Thus Yellin does not disclose nor even in any way suggest the last two clauses of claim 22.

Nagel discloses a method for modifying program executable code, possibly on an embedded system (col. 14, l. 56–66), possibly using a virtual machine (col. 11, l. 61–67). Nagel's disclosure is not otherwise relevant. Combining Nagel's disclosure with Yellin's would not have taught one of ordinary skill in the art the method of claim 22.

Claims 20, 24

Claims 20, 24 recite the same essential features as claim 4 and are thus patentable for the reasons explained above. Accordingly, Applicant respectfully requests withdrawal of the § 103 rejection of claims 20 and 24. Applicant also requests withdrawal of the § 103 rejection of claims 25-27 because the are dependent upon claim 24.

Secondary Considerations

Further, evidence shows that Applicant's invention is non-obvious. Yellin discloses a method of static verification of applet code during downloading of the applet. As cited in the specification, Yellin's method was also disclosed in "The Java Virtual Machine Specification", by Tim Lindholm and Frank Yellin (Addison-Wesley, 1996), attached to this response and available online at http://java.sun.com/docs/books/jvms/first_edition/html/VMSpecTOC.doc.html (see section 4.9). As mentioned in the specification (page 5, line 5 ff.), the method taught by Yellin does not appear to be suitable for embedded systems with limited memory capacity.

Sun Microsystems also suggested that Yellin's verifier is not suitable for embedded systems. Sun Microsystems, Inc., described a counterpart of Java called Java Card (see The Java Card 2.1 Virtual Machine Specification, published by Sun Microsystems, Inc. in June 1999, cited

in the specification; available online at http://java.sun.com/javacard/specs.html) which is more suitable for "smart cards and similar small-memory embedded devices" (see preface, p. xi, first paragraph). A portion of the specification is attached to this response. However, the Java Card specification specifically omits a verifier due to the perceived difficulty in supporting such a verifier on an embedded device such as a smart card (see §1.3, p. 4, first paragraph), and suggests instead to either perform a verification before loading the code onto the embedded device or performing only a subset of the verification. Applicant's method makes it possible to perform a fully effective verification on an embedded device.

Furthermore, the inventor published an article in 2002 titled "Bytecode verification on Java smart cards", in *Software: Practice & Experience* (32(4):319-340), a peer-reviewed journal. A copy of the article is attached to this response. The article appears to be based on the invention claimed in the subject application. The fact that a peer reviewed journal published an article about the invention goes to show that the invention was not considered to be known or obvious in the industry.

A declaration under 37 C.F.R. 1.132 accompanies this response. The declaration introduces the references listed above, namely, the Java Card 2.1 Specification, the Java Virtual Machine Specification, and the 2002 article.

G. Conclusion

In consideration of the amendments and foregoing discussion, the application is now believed to be in condition for allowance. Early allowance of the subject application is respectfully solicited. The Examiner is kindly invited to contact Applicants' agent at 617-305-2136 to facilitate prosecution.

This response should not require any additional fees. However, in the event that additional fees are due, please charge or credit any refund to our Deposit Account No. 50-2324.

Respectfully Submitted,

Dated: Sep. 8, 2009

/Seth A. Milman/
Seth A. Milman
Reg. No. 64,573

Holland & Knight LLP
10 St. James Avenue
Boston, MA 02116-3889
Telephone 617-305-2136
Facsimile 617-523-6850

8808655_v3